JavaOne℠

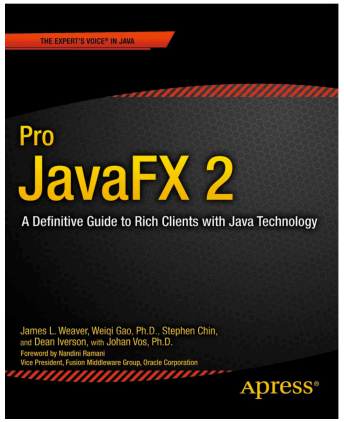
Hacking JavaFX with Groovy, Clojure, Scala, and Visage

Stephen Chin
Java Evangelist, Oracle
stephen.chin@oracle.com
tweet: @steveonjava

Meet the Presenter

Stephen Chin

- Java Evangelist, Oracle
- Author, Pro JavaFX Platform 2
- Open Source Hacker
  - JFXtras
  - ScalaFX
  - Visage
- User Group Co-Leader
  - Silicon Valley JavaFX User Group
  - Streamed Live!



Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

JavaFX 2.0 Platform

Immersive Application Experience

Leverage your Java skills with modern JavaFX APIs

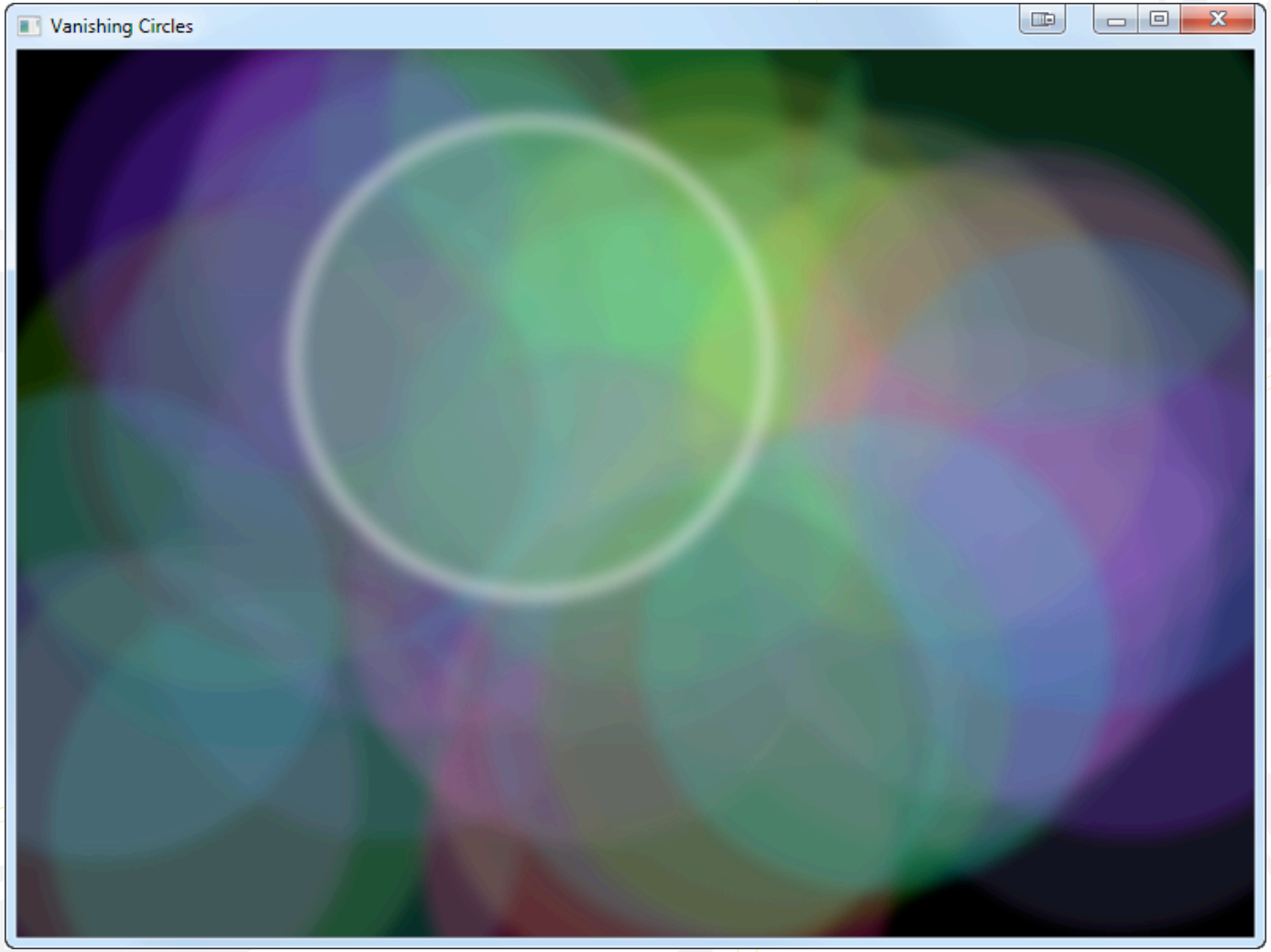
- Cross-platform Animation, Video, Charting
- Integrate Java, JavaScript, and HTML5 in the same application
- New graphics stack takes advantage of hardware acceleration for 2D and 3D applications
- Use your favorite IDE: NetBeans, Eclipse, IntelliJ, etc.

JavaFX With Java

Programming Languages

- JavaFX 2.0 APIs are now in Java
  - Pure Java APIs for all of JavaFX
  - Binding and Sequences exposed as Java APIs
  - FXML Markup for tooling
- Embrace all JVM languages
  - **Groovy**, **Scala**, **Clojure**, JRuby
  - Fantom, Mira, Gosu, Jython, etc.
- JavaFX Script is no longer supported by Oracle
  - Existing JavaFX Script based applications will continue to run
  - **Visage** is the open-source successor to the JavaFX Script language

Vanishing Circles



Vanishing Circles in Java

```
public class VanishingCircles extends Application {

  public static void main(String[] args) {
    Application.launch(args);
  }

  @Override
  public void start(Stage primaryStage) {
    primaryStage.setTitle("Vanishing Circles");
    Group root = new Group();
    Scene scene = new Scene(root, 800, 600, Color.BLACK);
    List<Circle> circles = new ArrayList<Circle>();
    for (int i = 0; i < 50; i++) {
      final Circle circle = new Circle(150);
      circle.setCenterX(Math.random() * 800);
      circle.setCenterY(Math.random() * 600);
      circle.setFill(new Color(Math.random(), Math.random(), Math.random(), .2));
      circle.setEffect(new BoxBlur(10, 10, 3));
      circle.addEventHandler(MouseEvent.MOUSE_CLICKED, new EventHandler<MouseEvent>() {
        public void handle(MouseEvent t) {
          KeyValue collapse = new KeyValue(circle.radiusProperty(), 0);
          new Timeline(new KeyFrame(Duration.seconds(3), collapse)).play();
        }
      });
      circle.setStroke(Color.WHITE);
      circle.strokeWidthProperty().bind(Bindings.when(circle.hoverProperty())
        .then(4)
        .otherwise(0));
      circles.add(circle);
    }
    root.getChildren().addAll(circles);
    primaryStage.setScene(scene);
    primaryStage.show();

    Timeline moveCircles = new Timeline();
    for (Circle circle : circles) {
      KeyValue moveX = new KeyValue(circle.centerXProperty(), Math.random() * 800);
      KeyValue moveY = new KeyValue(circle.centerYProperty(), Math.random() * 600);
      moveCircles.getKeyFrames().add(new KeyFrame(Duration.seconds(40), moveX, moveY));
    }
    moveCircles.play();
  }
}
```

40 Lines
1299 Characters

Application Skeleton

```
public class VanishingCircles extends Application {
  public static void main(String[] args) {
    Application.launch(args);
  }
  @Override
  public void start(Stage primaryStage) {
    primaryStage.setTitle("Vanishing Circles");
    Group root = new Group();
    Scene scene = new Scene(root, 800, 600, Color.BLACK);
    [create the circles…]
    root.getChildren().addAll(circles);
    primaryStage.setScene(scene);
    primaryStage.show();
    [begin the animation…]
  }
}
```

Create the Circles

```
List<Circle> circles = new ArrayList<Circle>();
for (int i = 0; i < 50; i++) {
  final Circle circle = new Circle(150);
  circle.setCenterX(Math.random() * 800);
  circle.setCenterY(Math.random() * 600);
  circle.setFill(new Color(Math.random(), Math.random(),
                           Math.random(), .2));
  circle.setEffect(new BoxBlur(10, 10, 3));
  circle.setStroke(Color.WHITE);
  [setup binding…]
  [setup event listeners…]
  circles.add(circle);
}
```

Setup Binding

```
circle.strokeWidthProperty().bind(Bindings
   .when(circle.hoverProperty())
   .then(4)
   .otherwise(0)
);
```

Setup Event Listeners

```
circle.addEventHandler(MouseEvent.MOUSE_CLICKED,
                       new EventHandler<MouseEvent>() {
  public void handle(MouseEvent t) {
    KeyValue collapse = new KeyValue(circle.radiusProperty(), 0);
    new Timeline(new KeyFrame(Duration.seconds(3),
                              collapse)).play();
  }
});
```

Begin the Animation

```
Timeline moveCircles = new Timeline();
for (Circle circle : circles) {
  KeyValue moveX = new KeyValue(circle.centerXProperty(),
                                Math.random() * 800);
  KeyValue moveY = new KeyValue(circle.centerYProperty(),
                                Math.random() * 600);
  moveCircles.getKeyFrames().add(new KeyFrame(Duration.seconds(40),
                                              moveX, moveY));
}
moveCircles.play();
```

JavaFX With Groovy

Features of Groovy

- Modern language
  - Closures
  - AST Transforms
  - Strongly typed dynamic language
- Tight integration with Java
  - Very easy to port from Java to Groovy
- Declarative syntax with GroovyFX Builders
  - Familiar to Groovy and JavaFX Script developers

Java vs. GroovyFX DSL

```
public class VanishingCircles extends Application {

  public static void main(String[] args) {
    Application.launch(args);
  }

  @Override
  public void start(Stage primaryStage) {
    primaryStage.setTitle("Vanishing Circles");
    Group root = new Group();
    Scene scene = new Scene(root, 800, 600, Color.BLACK);
    List<Circle> circles = new ArrayList<Circle>();
    for (int i = 0; i < 50; i++) {
      final Circle circle = new Circle(150);
      circle.setCenterX(Math.random() * 800);
      circle.setCenterY(Math.random() * 600);
      circle.setFill(new Color(Math.random(), Math.random(), Math.random(), .2));
      circle.setEffect(new BoxBlur(10, 10, 3));
      circle.addEventHandler(MouseEvent.MOUSE_CLICKED, new EventHandler<MouseEvent>() {
        public void handle(MouseEvent e) {
          KeyValue kv = new KeyValue(circle.opacityProperty(), 0);
          new Timeline(new KeyFrame(Duration.seconds(3), kv)).play();
        }
      });
      circle.setStroke(Color.WHITE);
      circle.strokeWidthProperty().bind(Bindings.when(circle.hoverProperty())
        .then(4)
        .otherwise(0));
      circles.add(circle);
    }
    root.getChildren().addAll(circles);
    primaryStage.setScene(scene);
    primaryStage.show();

    Timeline moveCircles = new Timeline();
    for (Circle circle : circles) {
      KeyValue moveX = new KeyValue(circle.centerXProperty(), Math.random() * 800);
      KeyValue moveY = new KeyValue(circle.centerYProperty(), Math.random() * 600);
      moveCircles.getKeyFrames().add(new KeyFrame(Duration.seconds(40), moveX, moveY));
    }
    moveCircles.play();
  }
}
```

40 Lines
1299 Characters

```
GroovyFX.start { primaryStage ->
  def sg = new SceneGraphBuilder()
  def rand = new Random().&nextInt
  def circles = []

  sg.stage(title: 'Vanishing Circles', show: true) {
    scene(fill: black, width: 800, height: 600) {
      50.times {
        circles << circle(centerX: rand(800), centerY: rand(600), radius: 150, stroke: white,
                strokeWidth: bind('hover', converter: {val -> val ? 4 : 0})) {
          fill rgb(rand(255), rand(255), rand(255), 0.2)
          effect b
          onMouseClicked { e ->
            timeli
              at(3
              }.play()
          }
        }
      }
    }

    timeline(cycle
      circles.each { circle ->
        at (40.s) {
          change circle.centerXProperty() to rand(800)
          change circle.centerYProperty() to rand(600)
        }
      }
    }.play()
  }
}
```

29 Lines
671 Characters

```
GroovyFX.start { primaryStage ->
  def sg = new SceneGraphBuilder()
  def rand = new Random().&nextInt
  def circles = []

  sg.stage(title: 'Vanishing Circles', show: true) {
    scene(fill: black, width: 800, height: 600) {
      50.times {
        circles << circle(centerX: rand(800), centerY: rand(600),
            radius: 150, stroke: white,
            strokeWidth: bind('hover', converter: {val -> val ? 4 : 0})) {
          fill rgb(rand(255), rand(255), rand(255), 0.2)
          effect boxBlur(width: 10, height: 10, iterations: 3)
        }
      }
    }
  }
}
```

```
GroovyFX.start { primaryStage ->
  def sg = new SceneGraphBuilder()
  def rand = new Random().&nextInt
  def circles = []

  sg.stage(title: 'Vanishing Circle
    scene(fill: black, width: 800,
      50.times {
        circles << circle(centerX: rand(800), centerY: rand(600),
            radius: 150, stroke: white,
            strokeWidth: bind('hover', converter: {val -> val ? 4 : 0})) {
          fill rgb(rand(255), rand(255), rand(255), 0.2)
          effect boxBlur(width: 10, height: 10, iterations: 3)
        }
      }
    }
  }
}
```

Builder for GroovyFX scene graphs

```
GroovyFX.start { primaryStage ->
  def sg = new SceneGraphBuilder()
  def rand = new Random().&nextInt
  def circles = []

  sg.stage(title: 'Vanishing Circles', show: true) {
    scene(fill: black, width: 800, height: 600) {
      50.times {
        circles << circle(centerX: ran
            radius: 150, stroke: white,
            strokeWidth: bind('hover',                    0})) {
          fill rgb(rand(255), rand(255), rand(255), 0.2)
          effect boxBlur(width: 10, height: 10, iterations: 3)
        }
      }
    }
  }
}
```

Declarative Stage definition

```
GroovyFX.start { primaryStage ->
  def sg = new SceneGraphBuilder()
  def rand = new Random().&nextInt
  def circles = []

  sg.stage(title: 'Vanishin
    scene(fill: black, width
      50.times {
        circles << circle(centerX: rand(800), centerY: rand(600),
            radius: 150, stroke: white,
            strokeWidth: bind('hover', converter: {val -> val ? 4 : 0})) {
          fill rgb(rand(255), rand(255), rand(255), 0.2)
          effect boxBlur(width: 10, height: 10, iterations: 3)
        }
      }
    }
  }
}
```

Inline property definitions

```
GroovyFX.start { primaryStage ->
  def sg = new SceneGraphBuilder()
  def rand = new Random().&nextInt
  def circles = []

  sg.stage(title: 'Vanishing Circles', show:
    scene(fill: black, width: 800, height: 6
      50.times {
        circles << circle(centerX: rand(800), centerY: rand(600),
            radius: 150, stroke: white,
            strokeWidth: bind('hover', converter: {val -> val ? 4 : 0})) {
          fill rgb(rand(255), rand(255), rand(255), 0.2)
          effect boxBlur(width: 10, height: 10, iterations: 3)
        }
      }
    }
  }
}
```

Bind to properties

```
GroovyFX.start { primaryStage ->
  def sg = new SceneGraphBuilder()
  def rand = new Random().&nextInt
  def circles = []

  sg.stage(title: 'Vanishing Circles', show
    scene(fill: black, width: 800, height:
      50.times {
        circles << circle(centerX: rand(800), centerY: rand(600),
            radius: 150, stroke: white,
            strokeWidth: bind('hover', converter: {val -> val ? 4 : 0})) {
          fill rgb(rand(255), rand(255), rand(255), 0.2)
          effect boxBlur(width: 10, height: 10, iterations: 3)
        }
      }
    }
  }
}
```

Sequence Creation Via Loop

Animation in GroovyFX

```
timeline(cycleCount: Timeline.INDEFINITE, autoReverse: true) {
  circles.each { circle ->
    at (40.s) {
      change circle.centerXProperty() to rand(800)
      change circle.centerYProperty() to rand(600)
    }
  }
}.play()
```

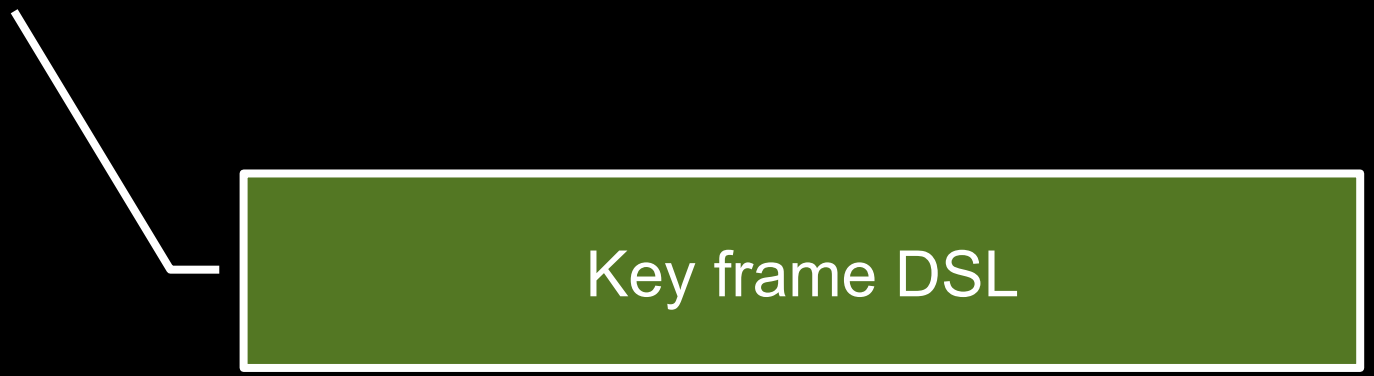
Animation in GroovyFX

```
timeline(cycleCount: Timeline.INDEFINITE, autoReverse: true) {
  circles.each { circle ->
    at (40.s) {
      change circle.centerXProperty() to rand(800)
      change circle.centerYProperty() to rand(600)
    }
  }
}.play()
```

Easy animation syntax:
at (duration) {keyframes}

Animation in GroovyFX

```
timeline(cycleCount: Timeline.INDEFINITE, autoReverse: true) {
  circles.each { circle ->
    at (40.s) {
      change circle.centerXProperty() to rand(800)
      change circle.centerYProperty() to rand(600)
    }
  }
}.play()
```



Animation in GroovyFX

```
timeline(cycleCount: Timeline.INDEFINITE, autoReverse: true) {
  circles.each { circle ->
    at (40.s) {
      change circle.centerXProperty() to rand(800) tween ease_both
      change circle.centerYProperty() to rand(600) tween linear
    }
  }
}.play()
```

Event Listeners in GroovyFX

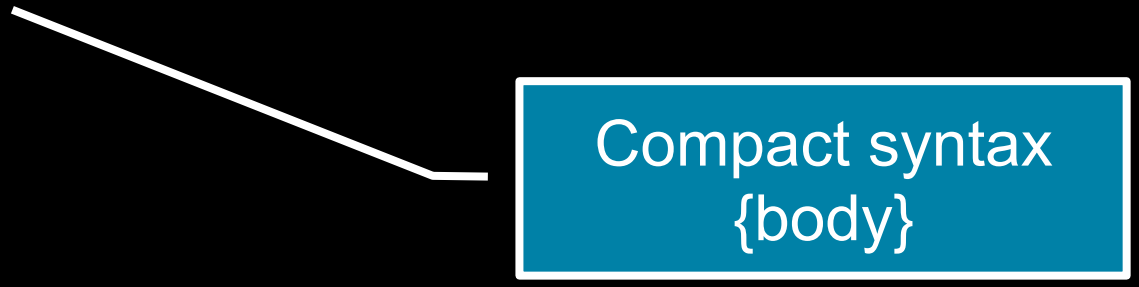
- Supported using the built-in Closure syntax
- Optional arguments for event objects

```
onMouseClicked { e ->
  timeline {
    at(3.s) { change e.source.radiusProperty() to 0 }
  }.play()
}
```

Event Listeners in GroovyFX

- Supported using the built-in Closure syntax
- Optional arguments for event objects

```
onMouseClicked { MouseEvent e ->
  timeline {
    at(3.s) { change e.source.radiusProperty() to 0 }
  }.play()
}
```



Event Listeners in GroovyFX

- Supported using the built-in Closure syntax
- Optional arguments for event objects

Optional event parameter
{event -> body}

```
onMouseClicked { MouseEvent e ->
  timeline {
    at(3.s) { change e.source.radiusProperty() to 0 }
  }.play()
}
```

But wait, there is more Grooviness…

Properties in Java

```
public class Person {
  private StringProperty firstName;
  public void setFirstName(String val) { firstNameProperty().set(val); }
  public String getFirstName() { return firstNameProperty().get(); }
  public StringProperty firstNameProperty() {
    if (firstName == null)
      firstName = new SimpleStringProperty(this, "firstName");
    return firstName;
  }

  private StringProperty lastName;
  public void setLastName(String value) { lastNameProperty().set(value); }
  public String getLastName() { return lastNameProperty().get(); }
  public StringProperty lastNameProperty() {
    if (lastName == null) // etc.
  }
}
```

Properties in GroovyFX

```
public class Person {
  @FXBindable String firstName;
  @FXBindable String lastName;
}
```

Properties in GroovyFX

```
public class Person {
   @FXBindable String firstName;
   @FXBindable String lastName = “Smith”;
}
```

TableView in Java

```
ObservableList<Person> items = ...
TableView<Person> tableView = new TableView<Person>(items);

TableColumn<Person,String> firstNameCol =
        new TableColumn<Person,String>("First Name");

firstNameCol.setCellValueFactory(
        new Callback<CellDataFeatures<Person, String>,
                     ObservableValue<String>>() {
  public ObservableValue<String> call(CellDataFeatures<Person, String> p)
  {
    return p.getValue().firstNameProperty();
  }
});

tableView.getColumns().add(firstNameCol);
```

TableView in GroovyFX

```
def dateFormat = new SimpleDateFormat("yyyy-MM-dd");

tableView(items: persons) {
  tableColumn property: "name",   text: "Name",   prefWidth: 150)
  tableColumn property: "age",    text: "Age",    prefWidth: 50)
  tableColumn property: "gender", text: "Gender", prefWidth: 150)
  tableColumn property: "dob",    text: "Birth",  prefWidth: 150,
              type: Date,
              converter: { from -> return dateFormat.format(from) })
}
```

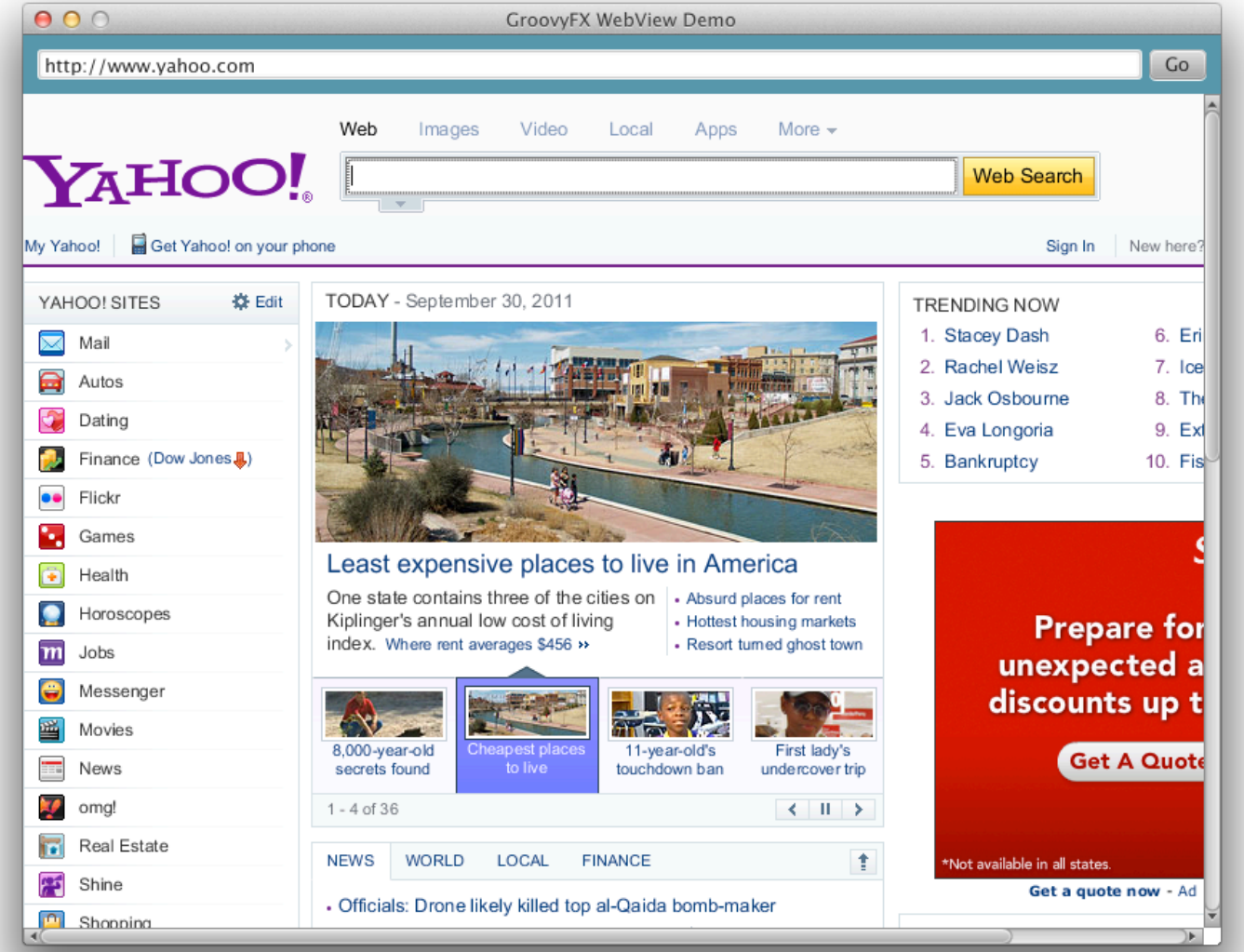
Layout in Java

```
TextField urlField = new TextField("http://www.google.com");
HBox.setHgrow(urlField, Priority.ALWAYS);

HBox hbox = new HBox();
hbox.getChildren().add(urlField);

WebView webView = new WebView();
VBox.setVgrow(webView, Priority.ALWAYS);

VBox vbox = new VBox();
vbox.getChildren().addAll(hbox, webView);
```

Layout in GroovyFX

```
sg.stage(title: "GroovyFX WebView Demo", show: true) {
    scene(fill: groovyblue, width: 1024, height: 800) {
        vbox {
            hbox(padding: 10, spacing: 5) {
                textField(“http://www.yahoo.com”, hgrow: "always")
                button("Go”)
            }
            webView vgrow: "always")
        }
    }
}
```

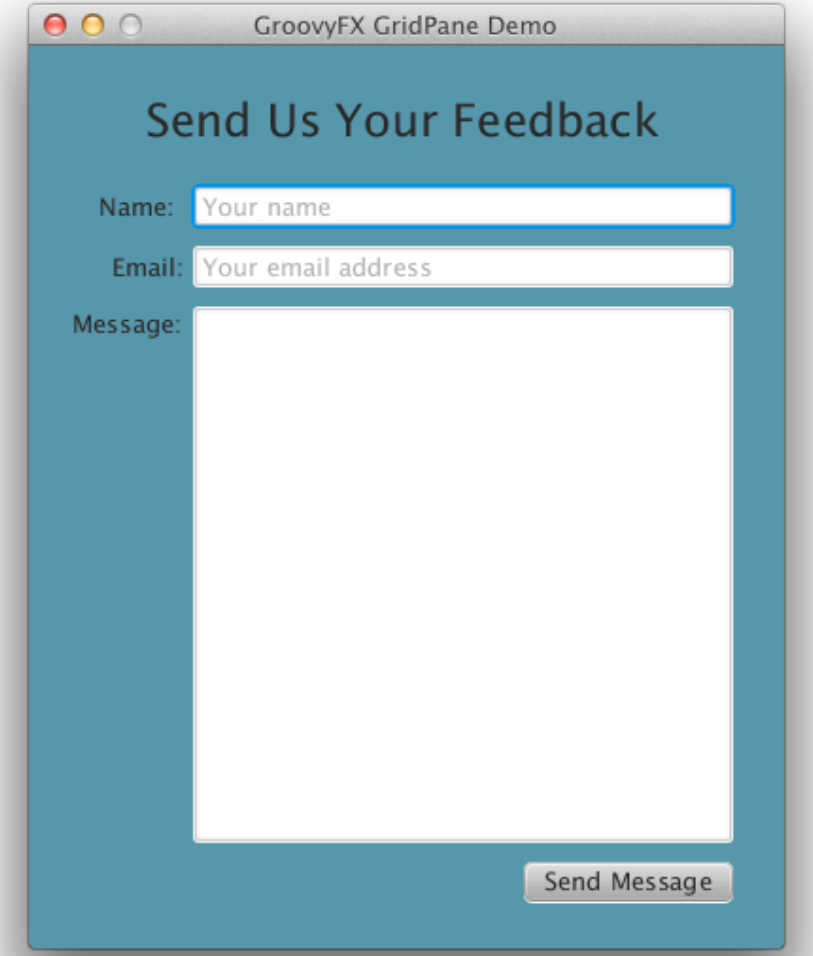
Layout in GroovyFX

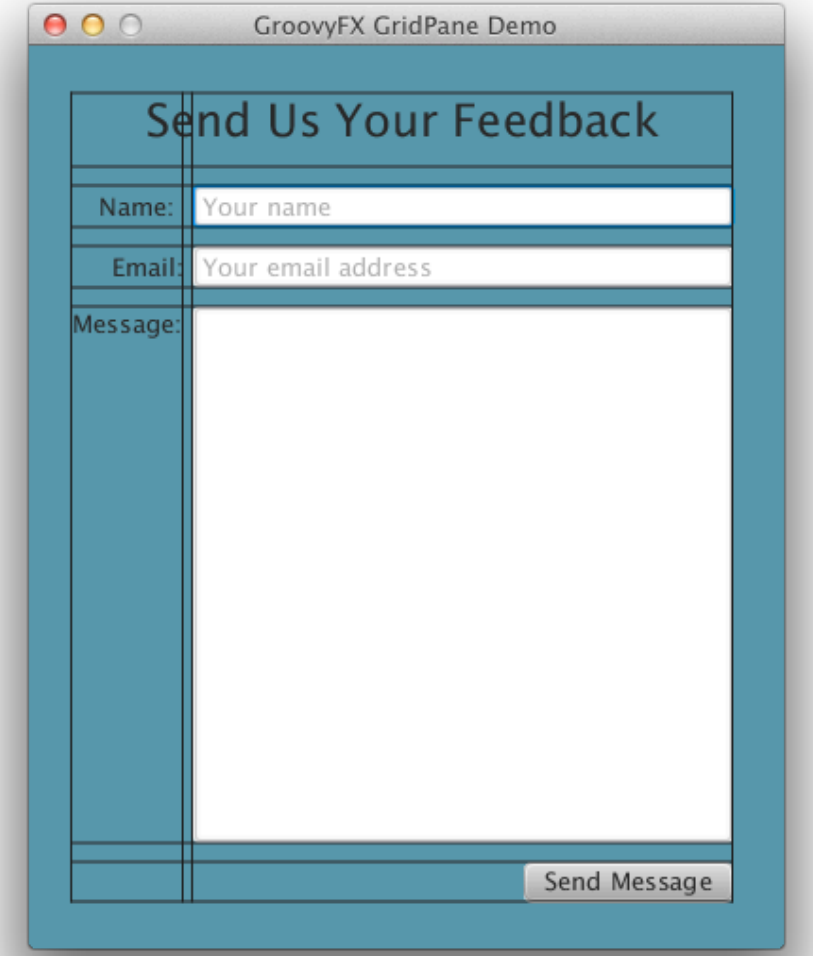


Layout in GroovyFX

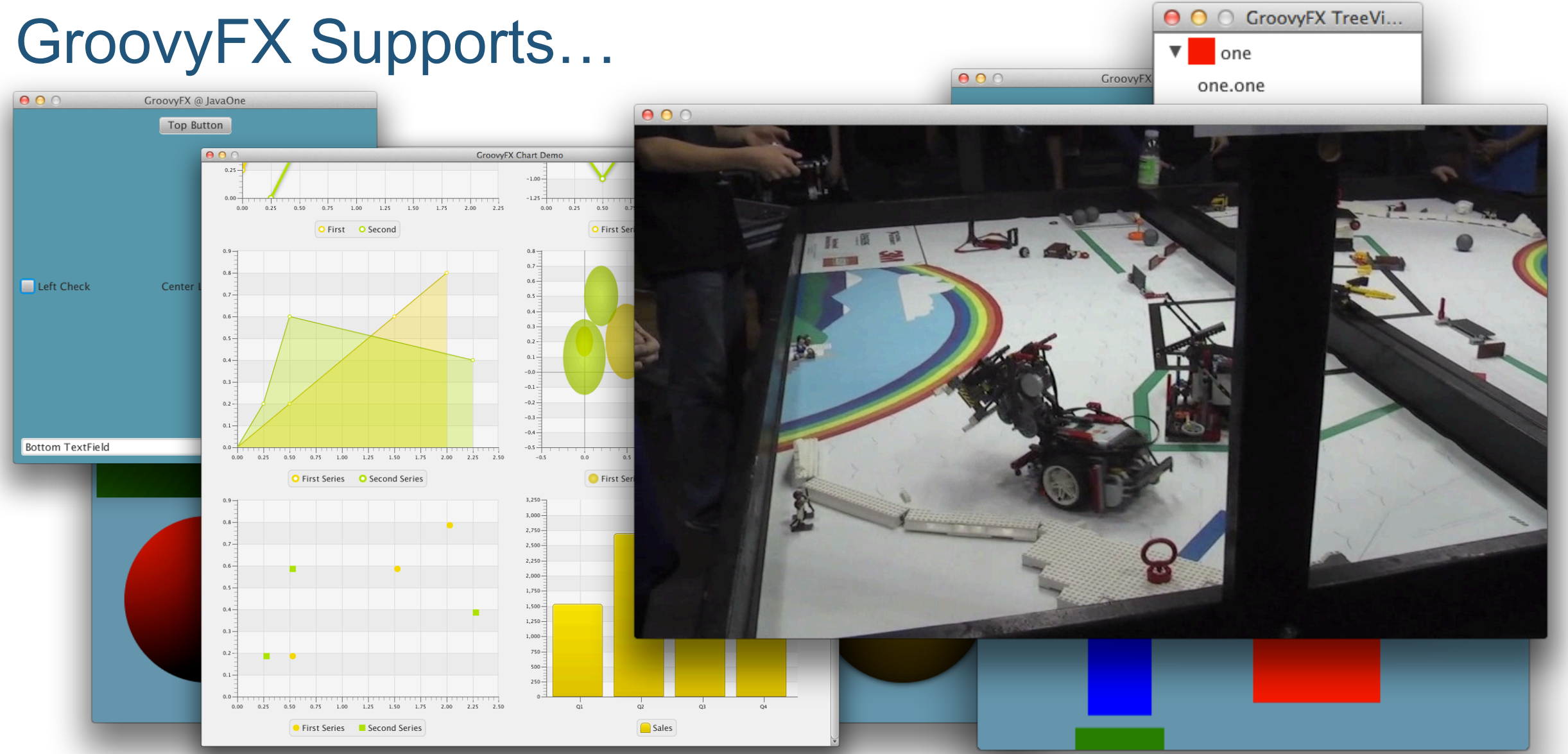
```
gridPane(hgap: 5, vgap: 10, padding: 25) {
  columnConstraints(minWidth: 50, halignment: "right")
  columnConstraints(prefWidth: 250)
  label("Send Us Your Feedback", font: "24pt sanserif",
        row: 0, columnSpan: GridPane.REMAINING, halignment: "center",
        margin: [0, 0, 10])

  label("Name: ", row: 1, column: 0)
  textField(promptText: "Your name", row: 1, column: 1, hgrow: 'always')

  label("Email:", row: 2, column: 0)
  textField(promptText: "Your email", row: 2, column: 1, hgrow: 'always')

  label("Message:", row: 3, column: 0, valignment: "baseline")
  textArea row: 3, column: 1, hgrow: "always", vgrow: "always")

  button("Send Message", row: 4, column: 1, halignment: "right")
}
```

Layout in GroovyFX

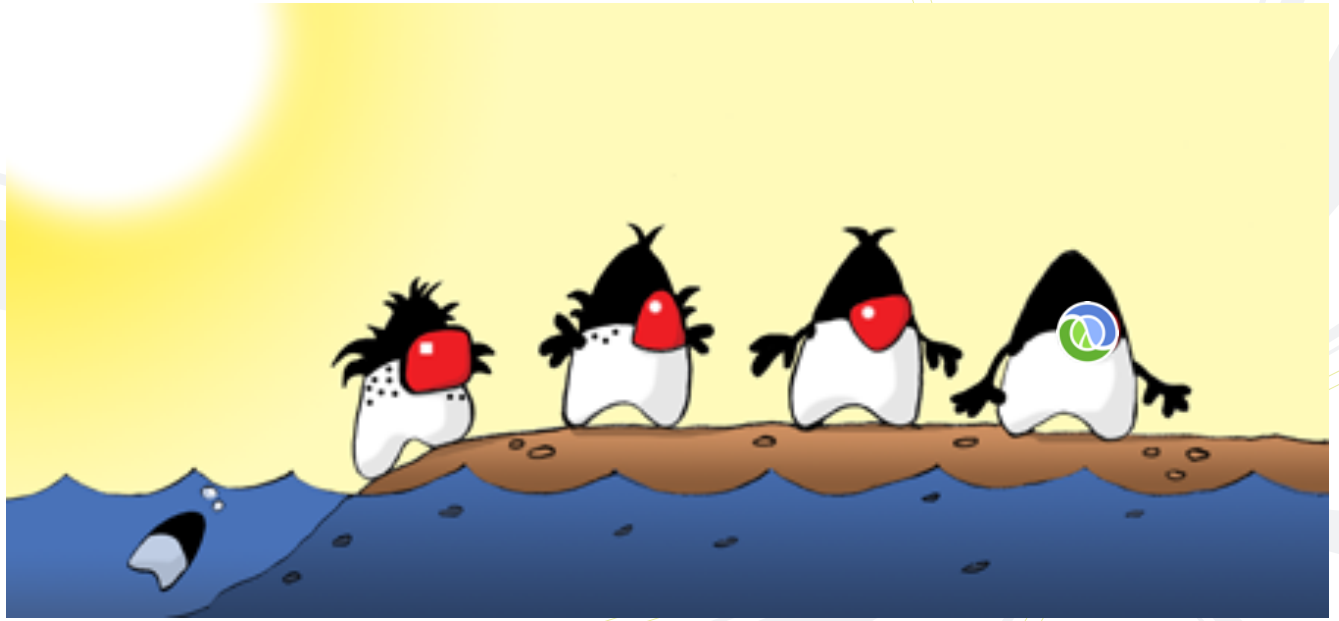




GroovyFX Supports…



JavaFX With Clojure



Artwork by Augusto Sellhorn http://sellmic.com/

A Little About Clojure

- Started in 2007 by Rich Hickey
- Functional Programming Language
- Derived from LISP
- Optimized for High Concurrency

```
(def hello (fn [] "Hello world"))
(hello)
```

- … and looks nothing like Java!

Clojure Syntax in One Slide

Symbols

- numbers – 2.178
- ratios – 355/113
- strings – “clojure”, “rocks”
- characters – \a \b \c \d
- symbols – a b c d
- keywords – :alpha :beta
- boolean – true, false
- null - nil

Collections

(commas optional)

- Lists

(1, 2, 3, 4, 5)

- Vectors

[1, 2, 3, 4, 5]

- Maps

{:a 1, :b 2, :c 3, :d 4}

- Sets

#{:a :b :c :d :e}

(plus macros that are syntactic sugar wrapping the above)

Clojure GUI Example

```
(defn javafxapp []
  (let [stage (Stage. "JavaFX Stage")
        scene (Scene.)]
    (.setFill scene Color/LIGHTGREEN)
    (.setWidth stage 600)
    (.setHeight stage 450)
    (.setScene stage scene)
    (.setVisible stage true)))
(javafxapp)
```

Refined Clojure GUI Example

```
(defn javafxapp []
  (doto (Stage. "JavaFX Stage")
    (.setWidth 600)
    (.setHeight 450)
    (.setScene (doto (Scene.)
      (.setFill Color/LIGHTGREEN)
      (.setContent (list (doto (Rectangle.)
        (.setX 25)
        (.setY 40)
        (.setWidth 100)
        (.setHeight 50)
        (.setFill Color/RED))))))
    (.setVisible true)))
(javafxapp)
```

Refined Clojure GUI Example

```
(defn javafxapp []
  (doto (Stage. "JavaFX Stage")
    (.setWidth 600)
    (.setHeight 450)
    (.setScene (doto (Scene.)
      (.setFill Color/LIGHTGREEN)
      (.setContent (list (doto (Rectangle.)
        (.setX 25)
        (.setY 40)
        (.setWidth 100)
        (.setHeight 50)
        (.setFill Color/RED))))))
    (.setVisible true)))
(javafxapp)
```

Doto allows nested data structures

Closures in Clojure

- Inner classes can be created using proxy

```
(.addListener hoverProperty
  (proxy [ChangeListener] []
    (handle [p, o, v]
      (.setFill rect
        (if (.isHover rect) Color/GREEN Color/RED)))))
```

Closures in Clojure

- Inner classes can be created using proxy

Proxy form:

```
(proxy [class] [args] fs+)
f => (name [params*] body)
```

```
(.addListener hoverProperty
  (proxy [ChangeListener] []
    (handle [p, o, v]
      (.setFill rect
        (if (.isHover rect) Color/GREEN Color/RED)))))
```

JavaFX With Scala

What is Scala

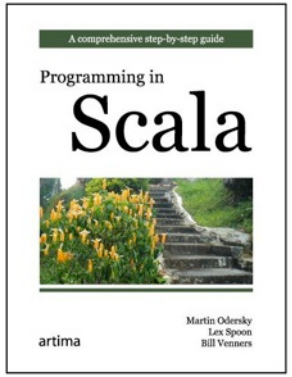
2001
- Scala Started

2003/2004
- Scala v1.0

2006
- Scala v2.0

2011
- Scala 2.9.2 (latest)

- Started in 2001 by Martin Odersky
- Compiles to Java bytecodes
- Pure object-oriented language
- Also a functional programming language



Java vs. Scala DSL

```
public class VanishingCircles extends Application {

  public static void main(String[] args) {
    Application.launch(args);
  }

  @Override
  public void start(Stage primaryStage) {
    primaryStage.setTitle("Vanishing Circles");
    Group root = new Group();
    Scene scene = new Scene(root, 800, 600, Color.BLACK);
    List<Circle> circles = new ArrayList<Circle>();
    for (int i = 0; i < 50; i++) {
      final Circle circle = new Circle(150);
      circle.setCenterX(Math.random() * 800);
      circle.setCenterY(Math.random() * 600);
      circle.setFill(new Color(Math.random(), Math.random(), Math.random(), .2));
      circle.setEffect(new BoxBlur(10, 10, 3));
      circle.addEventHandler(MouseEvent.MOUSE_CLICKED, new EventHandler<MouseEvent>() {
        public void handle(MouseEvent e) {
          KeyValue kv = new KeyValue(circle.radiusProperty(), 0);
          new Timeline(new KeyFrame(Duration.seconds(3), kv)).play();
        }
      });
      circle.setStroke(Color.WHITE);
      circle.strokeWidthProperty().bind(Bindings.when(circle.hoverProperty())
        .then(4)
        .otherwise(0));
      circles.add(circle);
    }
    root.getChildren().addAll(circles);
    primaryStage.setScene(scene);
    primaryStage.show();

    Timeline moveCircles = new Timeline();
    for (Circle circle : circles) {
      KeyValue moveX = new KeyValue(circle.centerXProperty(), Math.random() * 800);
      KeyValue moveY = new KeyValue(circle.centerYProperty(), Math.random() * 600);
      moveCircles.getKeyFrames().add(new KeyFrame(Duration.seconds(40), moveX, moveY));
    }
    moveCircles.play();
  }
}
```

40 Lines
1299 Characters

```
object VanishingCircles extends JFXApp {
  var circles: Seq[Circle] = null
  stage = new Stage {
    title = "Vanishing Circles"
    width = 800
    height = 600
    scene = new Scene {
      fill = BLACK
      circles = for (i <- 0 until 50) yield new Circle {
        centerX = random * 800
        centerY = random * 600
        radius = 150
        fill = color(random, random, random, .2)
        effect = new BoxBlur(10, 10, 3)
        strokeWidth = when(hover) then 4 otherwise 0
        stroke = WHITE
        onMouseClicked = {
          Timeline(at (3 s) {radius -> 0}).play()
        }
      }
      content = circles
    }
  }

  new Timeline {
    cycleCount = INDEFINITE
    autoReverse = true
    keyFrames = for (circle <- circles) yield at (40 s) {
      Set(
        circle.centerX -> random * stage.width,
        circle.centerY -> random * stage.height
      )
    }
  }.play();
}
```

33 Lines
591 Characters

```
object VanishingCircles extends JFXApp {
  stage = new Stage {
    title = "Disappearing Circles"
    width = 800
    height = 600
    scene = new Scene {
      fill = BLACK
      children = for (i <- 0 until 50) yield new Circle {
        centerX = random * 800
        centerY = random * 600
        radius = 150
        fill = color(random, random, random, 0.2)
        effect = new BoxBlur(10, 10, 3)
      }
    }
  }
}
```

```
object VanishingCircles extends JFXApp {
  stage = new Stage {
    title = "Disappearing Circles"
    width
    heigh
    scene
      fill = BLACK
      children = for (i <- 0 until 50) yield new Circle {
        centerX = random * 800
        centerY = random * 600
        radius = 150
        fill = color(random, random, random, 0.2)
        effect = new BoxBlur(10, 10, 3)
      }
    }
  }
}
```

Base class for JavaFX applications

```
object VanishingCircles extends JFXApp {
  stage = new Stage {
    title = "Disappearing Circles"
    width = 800
    height = 600
    scene = new Scene {
      fill = BLACK
      children = for (i <- 0 until 50) yield new Circle {
        centerX = random * 800
        centerY = random * 600
        radius = 150
        fill = color(random, random, random, 0.2)
        effect = new BoxBlur(10, 10, 3)
      }
    }
  }
}
```

Declarative Stage definition

```
object VanishingCircles extends JFXApp {
  stage = new Stage {
    title = "Disappearing Circles"
    width = 800
    height = 600
    scene = new Scene {
      fill = BLACK
      children = for (i <- 0 until 50) yield new Circle {
        centerX = random * 800
        centerY = random * 600
        radius = 150
        fill = color(random, random, random, 0.2)
        effect = new BoxBlur(10, 10, 3)
      }
    }
  }
}
```

Inline property definitions

```
object VanishingCircles extends JFXApp {
  stage = new Stage {
    title = "Disappearing Circles"
    width = 800
    height = 600
    scene = new Scene {
      fill = BLACK
      children = for (i <- 0 until 50) yield new Circle {
        centerX = random * 800
        centerY = random * 600
        radius = 150
        fill = color(random, random, random, 0.2)
        effect = new BoxBlur(10, 10, 3)
      }
    }
  }
}
```

Sequence Creation Via Loop

Binding in Scala

Infix Addition/Subtraction/Multiplication/Division:

```
height <== rect1.height + rect2.height
```

Aggregate Operators:

```
width <== max(rect1.width, rect2.width, rect3.width)
```

Conditional Expressions:

```
strokeWidth <== when (hover) then 4 otherwise 0
```

Compound Expressions:

```
text <== when (rect.hover || circle.hover && !disabled) then
  textField.text + " is enabled" otherwise "disabled"
```

Animation in Scala

```
val timeline = new Timeline {
  cycleCount = INDEFINITE
  autoReverse = true
  keyFrames = for (circle <- circles) yield at (40 s) {
    Set(
      circle.centerX -> random * stage.width,
      circle.centerY -> random * stage.height
    )
  }
}
timeline.play();
```

Animation in Scala

JavaFX Script-like animation syntax: at (duration) {keyframes}

```
val timeline = new Timeline {
  cycleCount = INDEFINITE
  autoReverse = true
  keyFrames = for (circle <- circles) yield at (40 s) {
    Set(
      circle.centerX -> random * stage.width,
      circle.centerY -> random * stage.height
    )
  }
}
timeline.play();
```

Animation in Scala

```
val timeline = new Timeline {
  cycleCount = INDEFINITE
  autoReverse = true
  keyFrames = for (circle <- circles) yield at (40 s) {
    Set(
      circle.centerX -> random * stage.width,
      circle.centerY -> random * stage.height
    )
  }
}
timeline.play();
```

Operator overloading for animation syntax

Animation in Scala

```
val timeline = new Timeline {
  cycleCount = INDEFINITE
  autoReverse = true
  keyFrames = for (circle <- circles) yield at (40 s) {
    Set(
 circle.centerX -> random * stage.width tween EASE_BOTH,
 circle.centerY -> random * stage.height tween EASE_IN
    )
  }
}
timeline.play();
```

Optional tween syntax

Event Listeners in Scala

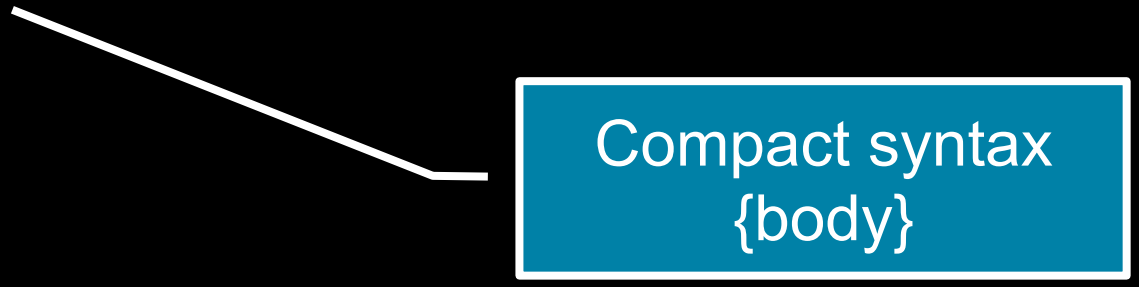
- Supported using the built-in Closure syntax
- Arguments for event objects
- 100% type-safe

```
onMouseClicked = { (e: MouseEvent) =>
  Timeline(at(3 s){radius->0}).play()
}
```

Event Listeners in Scala

- Supported using the built-in Closure syntax
- Arguments for event objects
- 100% type-safe

```
onMouseClicked = { (e: MouseEvent) =>
  Timeline(at(3 s){radius->0}).play()
}
```



Event Listeners in Scala

- Supported using the built-in Closure syntax
- Arguments for event objects
- 100% type-safe

Event parameter
{(event) => body}

```
onMouseClicked = { (e: MouseEvent) =>
  Timeline(at(3 s){radius->0}).play()
}
```

TableView in ScalaFX

```
def dateFormat = new SimpleDateFormat("yyyy-MM-dd")
new TableView[Speaker](persons) {
  columns = Seq(
    new TableColumn[Speaker, String] {
      text: "Name"
      converter = {_.firstName}
    } new TableColumn[Speaker, String] {
      text: "Age"
      converter = {_.age}
    } new TableColumn[Speaker, String] {
      text: "Gender"
      converter = {_.gender}
    } new TableColumn[Speaker, String] {
      text: "Birth"
      converter = {dateFormat.format(_.dob)},
    }
)}
```

JavaFX With Visage

About Project Visage

- ***"Visage is a domain specific language (DSL) designed for the express purpose of writing user interfaces."***

- Visage project goals:
  - Compile to JavaFX Java APIs
  - Evolve the Language (Annotations, Maps, etc.)
  - Support Other Toolkits

- Come join the team!
- For more info: http://visage-lang.org/

Java vs. Visage DSL

```
public class VanishingCircles extends Application {

  public static void main(String[] args) {
    Application.launch(args);
  }

  @Override
  public void start(Stage primaryStage) {
    primaryStage.setTitle("Vanishing Circles");
    Group root = new Group();
    Scene scene = new Scene(root, 800, 600, Color.BLACK);
    List<Circle> circles = new ArrayList<Circle>();
    for (int i = 0; i < 50; i++) {
      final Circle circle = new Circle(150);
      circle.setCenterX(Math.random() * 800);
      circle.setCenterY(Math.random() * 600);
      circle.setFill(new Color(Math.random(), Math.random(), Math.random(), .2));
      circle.setEffect(new BoxBlur(10, 10, 3));
      circle.addEventHandler(MouseEvent.MOUSE_CLICKED, new EventHandler<MouseEvent>() {
        public void handle(MouseEvent e) {
          KeyValue kv = new KeyValue(circle.radiusProperty(), 0);
          new Timeline(new KeyFrame(Duration.seconds(3), kv)).play();
        }
      });
      circle.setStroke(Color.WHITE);
      circle.strokeWidthProperty().bind(Bindings.when(circle.hoverProperty())
        .then(4)
        .otherwise(0));
      circles.add(circle);
    }
    root.getChildren().addAll(circles);
    primaryStage.setScene(scene);
    primaryStage.show();

    Timeline moveCircles = new Timeline();
    for (Circle circle : circles) {
      KeyValue moveX = new KeyValue(circle.centerXProperty(), Math.random() * 800);
      KeyValue moveY = new KeyValue(circle.centerYProperty(), Math.random() * 600);
      moveCircles.getKeyFrames().add(new KeyFrame(Duration.seconds(40), moveX, moveY));
    }
    moveCircles.play();
  }
}
```

40 Lines
1299 Characters

```
var circles:Circle[];
Stage {
  title: "Vanishing Circles"
  Scene {
    width: 800
    height: 600
    fill: BLACK
    Group {
      circles = for (i in [1..50]) {
        def c:Circle = Circle {
          centerX: random() * 800
          centerY: random() * 600
          radius: 150
          fill: Color.color(random(), random(), random(), .2)
          effect: BoxBlur {
            height: 10
            width: 10
            iterations: 3
          }
          stroke: bind if (c.hover) Color.WHITE else null
          strokeWidth: bind if (c.hover) 4 else 0
          onMouseClicked: function(e) {
            Timeline {at (3s) {c.radius => 0}}.play()
          }
        }
      }
    }
  }
}

Timeline {
  for (circle in circles) at (40s) {
    circle.centerX => random() * 800;
    circle.centerY => random() * 600
  }
}.play()
```

35 Lines
487 Characters

How about JavaFX on… Visage

```
Stage {
  title: "Vanishing Circles"
  scene: Scene {
    width: 800
    height: 600
    fill: BLACK
    content: Group {
      circles = for (i in [1..50]) {
        Circle {
          centerX: random() * 800
          centerY: random() * 600
        }
      }
    }
  }
}
```

How about JavaFX on… Visage

```
Stage {
  title: "Vanishing Circles"
  scene: Scene {
    width: 800
    height: 600
    fill: BLACK
    content: Group {
      circles = for (i in [1..50]) {
        Circle {
          centerX: random() * 800
          centerY: random() * 600
        }
      }
    }
  }
}
```

How about JavaFX on… Visage

```
Stage {
  title: "Vanishing Circles"
  Scene {
    width: 800
    height: 600
    fill: BLACK
    Group {
      circles = for (i in [1..50]) {
        Circle {
          centerX: random() * 800
          centerY: random() * 600
        }
      }
    }
  }
}
```

Visage is JavaFX Script++

- Default Parameters
- New Literal Syntax For:
  - Angles – `35deg, 4rad, 1turn`
  - Colors – `#DDCCBB, #AA33AA|CC`
  - Lengths – `5px, 2pt, 3in, 4sp`
- Null-check Dereference
  - var width = rect.!width
- Built-in Bindable Maps (coming soon!)
  - var fruitMap = ["red" : apple, "yellow" : banana]
  - var fruit = bind fruitMap["red"]

Visage and JavaFX 2.0 are made for each other…

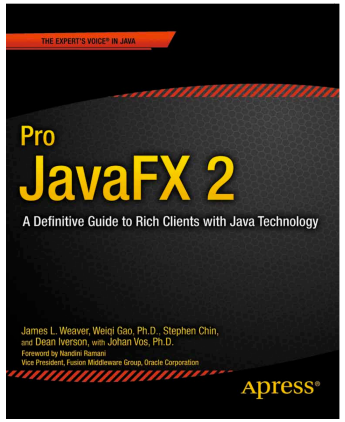
- Enhanced Binding
  - Retains lazy evaluation properties with additional expressive power
- Integrated Collections
  - Sequences and Maps automatically convert between JavaFX Observable Lists/Maps
- Built-in Animation Syntax
  - Ties into JavaFX animation subsystem
  - Provides consistent, clean APIs

Other JVM Languages to Try

- JRuby
  - Faithful to Ruby language with the power of the JVM
- Gosu
  - Up and coming language created at GuideWire
  - Easy to enhance libraries and create DSLs
- Mirah
  - Invented by Charles Nutter
  - Local Type Inference, Static and Dynamic Typing
- Fantom
  - Created by Brian and Andy Frank
  - Portable to Java and .NET
  - Local Type Inference, Static and Dynamic Typing

Conclusion

- You can write JavaFX applications in pure Java
- JavaFX is also usable in alternate languages
- You can get improved support using DSL libraries
  - GroovyFX
  - ScalaFX
- Or a dedicated UI JVM Language
  - Visage



Stephen Chin
stephen.chin@oracle.com
tweet: @steveonjava

Thanks to Dean Iverson and Jonathan Giles for help preparing this talk